AN INTRODUCTION TO THE HLA INTERFACE SPECIFICATION AND OBJECT MODEL TEMPLATE TEST PROCEDURES

Margaret L. Loper and David W. Roberts
Georgia Tech Research Institute
Georgia Institute of Technology
347 Ferst Drive
Atlanta, GA 30332
margaret.loper@gtri.gatech.edu
dw.roberts@gtri.gatech.edu

KEYWORDS
Testing, Test Procedures, HLA, Compliance

ABSTRACT

At the 14th DIS Workshop a position paper describing the HLA Test Process was presented. The paper described a two-phase test process consisting of HLA Compliance and Federation testing. Over the last six months, the test process has been refined based on knowledge from the HLA prototype federations. This paper will update the Test Process and discuss the test procedures being developed to support HLA Compliance Testing. In particular, the interface specification and object model template procedures will be introduced.

1.0 THE TEST PROCESS

The test process being developed to support HLA has two phases: HLA Compliance and Federation Testing. Within each phase several steps exist, as shown in Figure 1. It is important to note that the test process described here is closely linked and coordinated with the Verification, Validation, and Accreditation process (VV&A), that is the test process is part of VV&A.

The first phase of the test process is HLA Compliance. This phase consists of two steps: Federate and Federation compliance. The requirements for HLA compliance are described in the HLA Compliance Checklist [1]. The checklist is divided into three lists of items: Federate, Federation, and Run Time Infrastructure (RTI). In the test process, Federate Compliance refers to the Federate Checklist items and Federation Compliance refers to the Federation and RTI Checklist items.

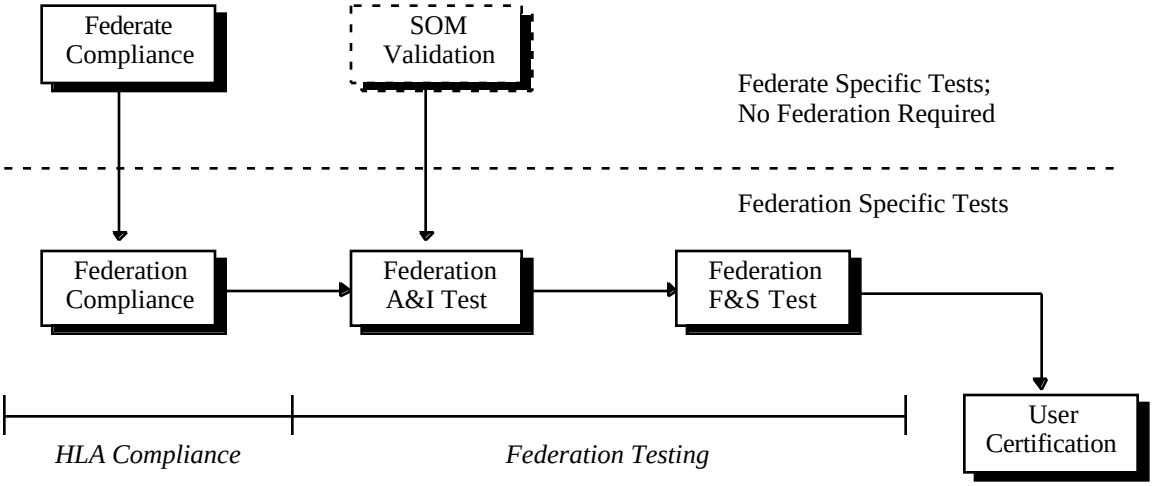

Figure 1: Test Process

The second phase of the test process is Federation Testing. This phase consists of two steps: Application & Integration (A&I) Testing, which addresses a Federation's ability to conform to Federation requirements, and Functional & Scenario (F&S) Testing which addresses compatibility among Federation members.

In HLA, all federates must have a Simulation Object Model (SOM) [2] in the Object Model Template (OMT) [3] format. A SOM documents a simulation's objects, their attributes, and interactions. In otherwords, the SOM documents a simulation's capabilities that can be used when building a federation. Determining whether a SOM is valid with respect to the simulation it is describing is an important consideration for Federation Testing.

1.1 HLA Compliance

As mentioned previously, the requirements for HLA Compliance are described in the HLA Compliance Checklist [1]. The Federate, Federation, and RTI checklist items are listed below. Procedures which support these items are describe in section 2.

1.1.1 Federate Items

1. Federates shall have an HLA SOM documented in accordance with the HLA OMT.
2. Federates shall be able to update and/or reflect any attributes of objects in their SOM and send and/or receive SOM object interactions externally, as specified in their SOM.
3. Federates shall be able to transfer and/or accept ownership of attributes dynamically during a federation execution, as specified in their SOM.
4. Federates shall be able to vary the conditions (e.g., thresholds) under which they provide updates of attributes of objects, as specified in their SOM.
5. Federates shall be able to manage local time in a way which will allow them to coordinate data exchange with other members of a federation.
6. During a federation execution, federates shall interact with the RTI in accordance with the HLA interface specification.

1.1.2 Federation Items

1. Federations shall have an HLA Federation Object Model (FOM), documented in accordance with the HLA OMT.
2. In a federation, all object representation shall be in the federates, not in the RTI.
3. During a federation execution, all exchange of FOM data among federates occur via the RTI.
4. During a federation execution, federates shall interact with the RTI in accordance with the HLA interface specification.
5. During a federation execution, an attribute of an instance of an object shall be owned by only one federate at any given time.

1.1.3 RTI Items

1. During a federation execution, the RTI shall interact with federates in accordance with the HLA interface specification.
2. The RTI shall provide services as called for by the federates via the interface in accordance with the RTI functional specification.
3. In a federation, all object representation shall be in the federates, not in the RTI.
4. During a federation execution, the RTI shall enforce the fact that an attribute of an instance of an object can be owned by only one federate at any given time.

1.2 Federation Testing

The purpose of federation testing is to ensure that the federation requirements are satisfied and that there is compatibility among simulations in a way that matters for the federation.

1.2.1 Application and Integration Testing

The purpose of A&I testing is threefold. First, testing must ensure that federates in the federation can interact via the RTI. Second, A&I testing must ensure that federation requirements specified in the FOM are satisfied. Federation requirements include issues such as data representations, interactions (and interaction protocols), and timing. Third, testing must ensure that specific federation agreements (e.g., paused save, ownership transfer) are satisfied.

1.2.2 Functional and Scenario Testing

The purpose of F&S testing is twofold. First, testing must ensure compatibility among federates in a way that matters for the federation. Compatibility issues include environment representation, object representation, fidelity, algorithms, and Conceptual Model of the Mission Space. Second, F&S testing must verify that a federation can perform scripted interactions and missions according to scenario

1.3 User Certification

The Federation User certifies the Federation Execution based on the results of the federation tests.

2.0 TEST PROCEDURES

There are two sets of test procedures being developed to support HLA Compliance testing: Interface Specification Test Procedures [5] and OMT Test Procedures [6]. This section will briefly describe these procedures. More detail about these procedures and conducting tests can be found in [5] and [6].

2.1 Interface Specification

The HLA Interface Specification [7] describes the interfaces between the federates and the RTI. The Interface Specification Test Procedures provide a standard set of procedures for this testing. The test procedures contain introductory and overview sections which describe:

- Definitions
- State Transition Diagrams
- Test Set-Up
- Test Methods
- Performing the Tests
- Assumptions

The actual test procedures are divided into three sections:

- Initialization and Control Services
- Action and Control Services
- Management Services

2.1.1 Execution Framework

The services contained in the Interface Specification are at an individual-level, each individual service is described in terms of its inputs, outputs and exceptions. There is no guidance or description for how the individual services are used together or the order in which they should be invoked. The test procedures provide this guidance by creating an execution framework (or recommended order) in which the interface services should be used. This is only guidance; there is no requirement to conform to this framework.

Test procedures are based on both individual-level and functional-level services. The test procedures define functional-level services as collections of individual-level services used for a specific function. Individual- and functional - level services are organized into four categories (Initialization, Control, Action, and Management), which form the execution framework as shown in Figure 2.

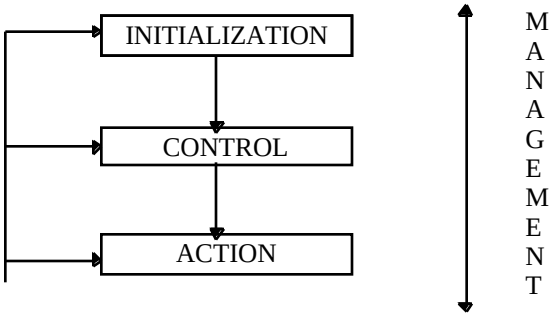

Figure 2: Test Procedures Execution Framework

Initialization Services are used by federates to set state in the RTI. These services typically follow a posting paradigm, in that the federate tells the RTI what it is capable of or what it requires for execution. Initialization services are at the individual-level and include: Create/Destroy Federation, Join/Resign Federation, Subscribe (Object Class, Object Attributes, Interaction Class), Publish (Object Class, Object Attributes, Interaction Class), Object Instantiation (Id Request, Instantiate Object), Initialize Federate Time and Rate, and Set Look Ahead.

Control Services are used by the RTI to initialize federates. These services also follow a posting paradigm in that the RTI tells the federate what publications and interactions it is allowed to update during the execution. Control services are also at an individual-level and include: Control Update, Control Interactions, and Instantiate Discovered Object.

Action Services are used by the RTI and federate to exchange information in order to accomplish an action. These services follow a request/response paradigm in that the federate requests some action to be completed and the RTI responds with confirmation or notification that the action has taken place. Action services are at the functional-level and include: Update/Reflect Attribute, Send/Receive Interaction, Request/Provide Attribute, Delete/Remove Object, Cancel Reflection/Remove Object, Attribute Ownership Acquisition, Attribute Ownership Divestiture, Unconditional Attribute Ownership Divestiture, Time Advance, Next Event, and Retract.

Management Services follow a request/response paradigm and are used to control the operation of the federation or to request information about the state of a federation. Management services can be invoked at any time after Initialization; all management services have a pre-condition that a Federation Execution exists and that the Federate is a member of the Execution. Management services are both individual-and functional-level services and include: Pause/Resume Federation Execution, Save/Restore Federation Execution (rolling), Save Federation Execution (paused), Query, Query Attribute Ownership, Change Federate Time, Change Federation Rate, Request Time, Request Rate, and Request Look Ahead.

2.1.2 State Transition Diagrams

The test procedures are based on state transition diagrams (STDs), which is a common method for specifying communication protocols. For each procedure, the individual- or functional-level services are defined in terms of a set of states, the inputs and their effect on the states, and the corresponding outputs. Thus, the state diagrams model the fact that a given input will cause the protocol entity (i.e., the part of the federate that implements the service) to go from one state to another, and possibly also cause a particular output to occur. The state diagrams are used in the test procedures purely for descriptive purposes: they serve as a model of reality.

2.1.3 Test Methods

Current interface testing is accomplished via black box testing. This means that the "Tester" cannot access the internal workings of the implementation of the Federate/Federation Under Test (FUT). Therefore, all the "Tester" can do is invoke services and see how the FUT reacts. This type of testing evaluates the order (and behavior) of services, it does not examine syntax and semantics. Currently, correct syntax is implied from successful interoperation with the RTI. Correct semantics is determined by invoking services according to the execution framework until such time a problem occurs. If a problem is found, the human "Tester" must consider which services are prerequisites for the service that failed and determine what is the likely problem. The black box testing approach described above is a generic process that can be used throughout the test process.

Another test method that can be used is white box testing. In this approach, the "Tester" has access to internal data from the FUT or can examine the FUT's code during testing. Since tools do not currently exist for this method of testing, a test process cannot be described. At some future date, the state variables described in the test procedures [5] may be used for observing the internal state of a FUT.

2.2 Object Model Template

The Object Model Template specifies the information content and format required for describing simulation and federation objects, attributes, and interactions. The OMT Test Procedures document is the first place that OMT test requirements have been documented other than in the text of the OMT documents themselves. As has been learned by the DIS community with DIS test procedures; specific, separate documentation of testable statements are necessary for testing. The OMT Test Procedures

provide a standard set of procedures for this testing.

The OMT Test Procedures contain sections for the following:

- Class Structures
- Object Interactions
- Attributes and Parameters
- Enumerated Data Types
- Lexicon
- Component Structures
- Associations
- Object Model Metadata

Each of these sections tracks with a section from the OMT [3] and OMT Extensions [4] documents. Similar to the Interface test procedures, the OMT test procedures define the tables that are required for both SOMs and FOMs, and in some instances, the allowed set of information which can be used to populate those tables. The OMT does not require procedures for testing tables, rather it requires procedures for ensuring consistency across tables.

2.2.1 Test Methods

Since the process for developing SOMs and FOMs is now a manual process, testing is also manual. This consists of a person verifying that there is consistency across tables entries and verifying that appropriate information is entered into SOM vs FOM tables. Currently, OMT development tools are being prototyped to produce SOMs and FOMs. It is expected that these tools will be tested to verify conformance with the OMT, thus facilitating automated testing.

3.0 LESSONS LEARNED FROM PROTOFEDERATIONS

Due to the aggressive schedules of the HLA prototype federations (or protofederations), many lessons learned from informal testing have not been received as of the writing of this paper. However, feedback that has been received suggests that the Interface test procedures have been very useful in two ways. First, the procedures begin to bridge-the-gap between the specification of the services and the use of the services. Specifically, the execution framework provides helpful information on ordering the Interface services. Second, the interface procedures are being used as a guide from which federations can develop their test plans. Instead of following the Interface test procedures exactly as described in [5], protofederations extracted parts of the procedures and used them in the context of a larger "integration" test plan.

The OMT test procedures were developed after the protofederations developed their FOMs. Therefore, we have received very little feedback regarding their use in testing.

4.0 NEXT STEPS

The majority of the work accomplished to date has focused on the HLA Compliance phase of the test process. Procedures have been developed which support the HLA Compliance Checklist and we are gaining knowledge on how those procedures should evolve. The next step is to conduct "formal" federate and federation testing such that experience can be gained in test tools.

The part of the test process that requires further work is Federation Testing. While the DIS community has experience in A&I testing, very little has been done for F&S testing. Studies and analyses will be required to better understand the range of tests required for functional testing and the types of tools needed to support these tests.

5.0 REFERENCES

The most current versions of the referenced documents can be found on the DMSO web page at http://www.dmso.mil.

[1] Defense Modeling and Simulation Office, High Level Architecture Compliance Checklist, 12 July 1996.
[2] Defense Modeling and Simulation Office, High Level Architecture Rules, Version 1.0, 9 August 1996.
[3] Defense Modeling and Simulation Office, High Level Architecture Object Model Template, Version 1.0, 7 August 1996.
[4] Defense Modeling and Simulation Office, High Level Architecture OMT Extensions, Version 0.3, 23 May 1996.
[5] Defense Modeling and Simulation Office, High Level Architecture Interface Specification Test Procedures, Version 0.2, 4 May 1996.
[6] Defense Modeling and Simulation Office, High Level Architecture Object Model Template Test Procedures, Version 0.1, 9 July 1996.
[7] Defense Modeling and Simulation Office, High Level Architecture Interface Specification, Version 1.0, 6 August 1996.